A New Approach to Threshold Attribute Based Signatures

S Sharmila Deva Selvi, Subhashini Venugopalan, C. Pandu Rangan

Theoretical Computer Science Laboratory
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Abstract

Inspired by developments in attribute based encryption and signatures, there has recently been a spurt of progress in the direction of threshold attribute based signatures (t-ABS). In this work we propose a novel approach to construct threshold attribute based signatures inspired by ring signatures. Threshold attribute based signatures, defined by a (t, n^*) threshold predicate, ensure that the signer holds atleast t out of a specified set of n^* attributes to pass the verification. Another way to look at this would be that, the signer has atleast 1 out of the $\binom{n^*}{t}$ combination of attribute sets. Thus, a new approach to t-ABS would be to let the signer pick some n' sets of t attributes each, from the $\binom{n^*}{t}$ possible sets, and prove that (s)he has atleast one of the n' sets in his/her possession. In this work, we provide a flexible threshold-ABS scheme that realizes this approach. We also prove our scheme to be secure with the help of random oracles.

Keywords

attribute-based, signature, threshold, t-ABS

1 Introduction

Attribute based signatures(ABS) is a cryptographic primitive in which users produce signatures based on some predicate of attributes, using keys issued by one or more attribute authorities. ABS has largely been inspired by attribute based encryption schemes [6, 1, 18]. Attribute based systems are applicable in settings where there is a need for a complex policy to govern the access of a document or provide authentication. These systems are also privacy-friendly since they deal with the attributes of a user and not with any direct identity that is associated with the signer. In this sense, ABS is similar to signature variants like Group signatures [3], Ring signatures [15] and Mesh signatures [2]. The dominant idea of all these signature primitives is that they allow the signer fine-grained control over the amount of personal information exposed. However, it is important to note that a valid ABS signature guarantees that only a person possessing the required attributes to satisfy the predicate can produce the signature.

A notable feature of ABS is that, unlike other signature schemes attribute based systems are capable of supporting complex predicate policies. For instance, some permissions can be approved only by a person who is: (((Major) AND (in Army OR Navy)) OR (Captain AND in Operation-Star) OR (Commander AND in Operation-X)). Moreover, a valid signature based on the above predicate would only indicate one of the four possibilities for the signer: *a*) Major in Army or *b*) Major in Navy or *c*) Captain in Operation-Star or *d*) Commander in Operation-X; but it would not reveal which of these the signer actually is. Also, a person who is not any of the four would not be able to produce a valid signature.

Collusion resistance is an important property of attribute based systems. This essentially means that multiple parties cannot collude and combine all their attributes to produce a valid signature if any one party could not do it individually. For example, a commander in the airforce and a captain in operation-X should not be able to somehow combine their attributes to produce a valid signature for the example predicate.

Threshold ABS is an attribute based signature where the predicate (or the signing policy) is given as a (t, n^*)-threshold over a verification attribute set of cardinality n. The verification of the signature ensures that the signer has a threshold number of, atleast t, attributes in common with the verification attribute set.

1.1 Related work

Work on attribute based signatures was influenced primarily by attribute based encryption primitives. Attribute based encryption took its form from Sahai and Waters work in [16], and was later formalized by Goyal *et al.* in [6] in 2006. The first formal notion of attribute based signatures was presented in 2008 in the

work of Maji *et al.* in [13]. Since then there are many ABS schemes that have come into literature. Maji *et al.*'s ABS scheme supported predicates having AND, OR and threshold gates, but the security of their scheme is in the generic group model. Earlier, Khader in [9, 8] introduced attribute based group signatures, which find applicability in settings where one might want additional credentials from a member of a group and also insist on collusion resistance.

Li and Kim in [12] introduced the notion of attribute based ring signatures. They have developed two schemes which they have shown to be secure based on the standard computational Diffie-Hellman assumption. However, their scheme is restricted to the selective unforgeability model. In addition, their ABS schemes support only those predicates with conjunctions, i.e. an (n,n) threshold, therefore, even though the signer's identity remains concealed, the exact attributes of the signer are exposed. Shahandashti and Safavi-Naini more concretely formalized the notions of threshold-ABS and its properties in [17]. Here, they not only gave the definitions and features of t-ABS schemes but they made several improvements on [12] and gave ABS schemes supporting a (k,n) threshold. Here k indicates the threshold and n the number of attributes in the verification attribute set. Although this scheme was more expressive, it was not efficient since the message was signed separately using each and every secret attribute possessed by the user. More recently, Li *et al.* in [11] propose two new and efficient threshold ABS schemes, one claimed to be secure in the random oracle model and the other in the standard model. The work in [11] is extended by Kumar *et al.* in [10] to construct a bounded multi-level threshold ABS scheme. Also, another recent work of Maji *et al.* in [14] gives a general framework for the construction of ABS schemes and some practical instantiations based on standard assumptions. Their scheme uses monotone span programs to incorporate the access structure and also make use of non-interactive witness indistinguishability (NIWI) to add to the anonymity of the signer. The authors also introduce a new generic primitive called credential bundle which is used in the key generating phase to bundle the attributes of the signer; this helps in making their scheme collusion resistant.

1.2 Our Contribution

This work provides a new perspective to threshold attribute based signatures using the ring concept. Any threshold attribute based signature ensures that the signer possesses atleast t out of the specified signing attributes, say n^* in number. From an other perspective, this is equivalent to saying that the signer has atleast 1 out of the $\binom{n^*}{t}$ combination of the attribute sets. So, in our scheme we let the signer pick some n' sets of t attributes each from the $\binom{n^*}{t}$ possible sets, and prove, using a ring signature, that (s)he has atleast one of the n' sets in his/her possession. Our scheme is proved secure by reduction to the modified CBDH assumption with the help of random oracles. We show both, unforgeability of the signature as well as the anonymity of the attribute-set used in signing.

We also show that our approach to t-ABS provides an interesting way for the signer to control privacy in situations where the signing policy (threshold predicate) is determined by an authority other than the signer. Additionally, our scheme can provide a constant-size signature if the signer chooses to show the exact attribute set (s)he uses for the signature. However, if attribute privacy is a strongly desired property then our scheme can be used to give signatures that can provide a balance between signer's attribute privacy and the size (and number of components) in the signature.

Organization of the paper. We begin with the preliminaries in the next section. Then, in Section 3, we present the construction of a t-ABS scheme in the novel approach we have just mentioned. This is followed by the proof of security of the scheme in the random oracle model.

2 Preliminaries

In this section we present some of the preliminaries required and also the construction of the efficient threshold ABS scheme by Li et al [11].

2.1 Bilinear Pairing

Let $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ be multiplicative groups of prime order p. The elements $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. A bilinear pairing is a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to G_T$ with the following properties:

1. **Bilinear**: $e({g_1}^a, {g_2}^b) = e(g_1, g_2)^{ab}$ for all $g_1 \in \mathbb{G}_1$,$g_2 \in \mathbb{G}_2$, where $a, b \in \mathbb{Z}_p$.
2. **Non-degenerate**: There exists $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ such that $e(g_1, g_2) \neq 1$; in other words, the map does not send all pairs in $\mathbb{G}_1 \times \mathbb{G}_2$ to the identity in $\mathbb{G}_T$.
3. **Computability**: There is an efficient algorithm to compute $e(g_1, g_2)$ for all $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$.

2.2 Lagrange Interpolation

Let $q(x)$ be a $d-1$ degree polynomial with each co-efficients in $\mathbb{Z}_p$. Then, given any set of d points on the polynomial $\{q(i) : i \in S\}$, where S is a set of indices such that $|S| = d$, we can use Lagrange's interpolation to find $q(j)$ for any $j \in \mathbb{Z}_p$ as follows: ($\Delta_{i,S}(j)$ is termed the Lagrange coefficient)

$$q(j) = \sum_{i \in S} q(i)\Delta_{i,S}(j) \text{ , where } \Delta_{i,S}(j) = \prod_{j' \in S, j' \neq i} \frac{j - j'}{i - j'}$$

2.3 Modified Computational Bilinear Diffie-Hellman Assumption.

We'll state the modified computational bilinear Diffie-Hellman problem [5], as we use it, to prove the security of our scheme.

Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be an efficiently computable bilinear map, where $\mathbb{G}$ has prime order p. The modified computational bilinear diffie-hellman(m-CBDH) assumption is said to hold in $\mathbb{G}$ if, given elements $\{P, aP, bP, cP, a^{-1}P\}$, then no probabilistic polynomial-time adversary can compute $e(P, P)^{abc}$ with non-negligible advantage, where $a, b, c \in_R \mathbb{Z}_p^*$ and generator $P \in \mathbb{G}$ are chosen independently and uniformly at random.

2.4 Forking Lemma

We make use of the forking lemma to give the proof of unforgeability of the threshold attribute based signature that we propose in Section3. Here, we will first present the conditions that are necessary for a ring signature to be considered *generic* and then define the forking lemma for generic ring signatures. The definitions are borrowed from those given by Herranz *et al.* in [7].

Generic Ring Signature. We denote by $H(\cdot)$, a cryptographic hash function that outputs k bits, where k is the security parameter. Consider a group of n ring members. Now, given the input message m, a generic ring signature scheme produces a tuple $(m, R_1, \cdots, R_n, h_1, \cdots, h_n, \sigma)$, where $R_1, \cdots, R_n$ (randomness) take their values randomly in a large set G in such a way that $R_i \neq R_j$ for all $i \neq j$, h_i is the hash value of (m, R_i), for $1 \leq i \leq n$, and the value σ is fully determined by $R_1, \cdots, R_n, h_1, \cdots, h_n$ and the message m.

Another required condition is that no R_i can appear with probability greater than $2/2^k$, where k is the security parameter. This condition can be achieved by choosing the set G as large as necessary.

Theorem 2.1 (Forking lemma) *The forking lemma for adaptive chosen message attacks with respect to generic ring signature schemes, as given in [7] is as follows. Let* $\mathbf{A}$ *be a probabilistic polynomial time Turing machine whose input only consists of public data. We denote by* q_h *and* q_s*, the number of queries that* $\mathbf{A}$ *can ask to the random oracle and to some real signers of the ring, respectively; we also denote by* $P_{q_h,n}$ *the number of n-permutations of* q_h *elements, i.e* $P_{q_h,n} = q_h(q_h - 1) \cdots (q_h - n + 1)$*. Assume that, within time bound* T*,* $\mathbf{A}$ *produces with non-negligible probability* ϵ*, a valid ring signature* $(m, R_1, R_2, \cdots, R_n, h_1, \cdots h_n, \sigma)$*. Suppose, the valid ring signature can be simulated with a polynomially indistinguishable distribution of probability, without knowing any of the secret keys of the ring , within a time bound of* T_s*. Then there exists another probabilistic polynomial time Turing machine which can, by a replay of attacker* $\mathbf{A}$ *where the interactions with the signer are simulated, produce two valid ring signatures* $(m, R_1, R_2, \cdots, R_n, h_1, \cdots h_n, \sigma)$ *and* $(m, R_1, R_2, \cdots, R_n, h'_1, \cdots h'_n, \sigma')$ *such that* $h_j \neq h'_j$*, for some* $j \in \{1, \cdots, n\}$ *and* $h_i = h'_i$ *for all* $i = 1, \cdots, n$ *such that* $i \neq j$*, in expected time* $\mathsf{T}' \leq \frac{144823(P_{q_h,n})(\mathsf{T}+q_s\mathsf{T}_s)}{\epsilon}$ *with non-negligible probability.*

2.5 Attribute-Based Signature

Attribute based signature schemes consist of four algorithms: setup, key generation, signing and verification algorithms, defined as follows:

Setup is run by a central authority. It takes as input the security parameter (d), and gives as output the set of public parameters denoted by *params* and a master secret key, *msk*.

Key-Gen (or **Extract**) is run by the key generating authority. It takes as input *msk* and a set of attributes (from the user/signer) to produce $\mathcal{D}$, a corresponding set of private keys for the signer.

Sign algorithm takes the signer's private keys $\mathcal{D}$, a predicate Υ, and a message m to produce a signature σ.

Verify algorithm ensures that, a signature σ on a message m is pronounced as valid *if and only if* the signer possesses attributes that satisfy the predicate Υ.

2.6 Threshold Attribute-Based Signature

Inorder to realize threshold-attribute signatures in this new setting, we propose the following model and security game.

Definition: This scheme consists of the following four algorithms:

Setup is run by a central authority. It takes as input the security parameter (d), and gives as output the set of public parameters denoted by $params$ and a master secret key, msk.

Key-Gen (or **Extract**) is run by the key generating authority. It takes as input msk and a set of signer attributes (U_β) to produce a corresponding set of private keys, $\mathcal{D}$, for the signer.

Sign algorithm takes the signer's private keys $\mathcal{D}$, a fixed threshold t (integer), an attribute subset U^*, and a message m to produce a signature σ. The *Sign* algorithm also outputs (as part of the signature), a set T consisting of n' subsets ($1 \leq n' \leq \binom{n^*}{t}$) of attributes T_i ($T = \{T_1, T_2, \cdots, T_{n'}\}$) such that each T_i satisfies $|T_i| = t$ and $T_i \subseteq U^*$ i.e. every subset T_i has threshold number of attributes and all of these attributes are present in U^* .

Verify algorithm is run by a verifier. It outputs 1 when a signature σ on a message m is valid i.e if one of the attribute subsets T_i was used in obtaining the signature.

Correctness: A threshold-ABS scheme must satisfy the correctness property, i.e. a signature generated by a signer with attribute set U_β must pass the verification test for the given U^* and t if $|U_\beta \cap U^*| \geq t$. More precisely, there exists an attribute subset $T_j \in T$ such that $T_j \subseteq U_\beta$. (and by definition $|T_j| = t$ and hence $|U_\beta \cap U^*| \geq t$)

Unforgeablility: It is required for the above threshold attribute based signature scheme to be existentially unforgeable under chosen attribute and message attacks as follows:

Setup phase: The challenger $\mathcal{C}$ runs the Setup algorithm and gives the common public parameters $params$ to adversary $\mathcal{A}$.

Query phase: Adversary can perform polynomially bounded number of queries in an adaptive manner (interactively) as described below.

- Hash. Adversary is allowed to query all the hash functions.
- KeyGen. $\mathcal{A}$ is allowed to query the keys for any set of attributes U_β.
- Sign. $\mathcal{A}$ requests for the signature of a signer with (any) attribute set U_β on any message m by specifying a threshold t' on an $n-$element attribute set U^*.

Forgery phase: At the end of the game, $\mathcal{A}$ outputs a forgery σ with respect to the set of attributes U^*, threshold t'' and message m' where σ contains n' attribute subsets T_i i.e.$T_1, T_2, \cdots, T_{n'}$ such that $\forall T_i$, $|T_i| = t''$, $T_i \subseteq U'$ and $1 \leq n' \leq \binom{n^*}{t}$.

$\mathcal{A}$ wins the game if the following hold:

1. σ is a valid signature with respect to U^*, t'', m'.
2. for all *KeyGen* queries on attribute set U_β, we have $|U_\beta \cap U^*| < t''$.
3. for all *Sign* queries on message m using signer attributes U_β, $m \neq m'$ or $|U_\beta \cap U^*| < t''$.

If no polynomial adversary has a considerable advantage in the above game, we say that the threshold-ARBS scheme is existentially unforgeable against chosen attribute and message attacks.

Collusion-resistance. Note here that the above game ensures the property *collusion resistance*. This is because the adversary could have queried for the secret keys of all elements in U' from *KeyGen* with different attribute sets U_β as input, such that $U' \subseteq \bigcup_{\forall \beta} U_\beta$. Thus, the game guarantees that no colluding group of users can create a signature that could not have been created by any one of the colluders independently.

Weak signer attribute privacy. This is an additional property that some threshold-ABS schemes provide. It says that the threshold attribute based signature does not reveal any information about the attributes of the signer other than saying the signer has t of the attributes in U^*. In our t-ARBS scheme, we define the scheme to have weak signer attribute privacy if the threshold attribute based signature does not reveal any information about the attributes of the signer except those attributes that the signer chooses to reveal in the n' chosen subsets (of the signing policy attribute set U^*) . To be more specific, given the n' subsets of attributes, the verifier must be unable to deduce which subset was used by the signer in order to give the signature.

3 New t-ABS Scheme

In this section we propose a novel scheme for threshold attribute based signatures which is conceptually based on ring signatures. Here is a brief sketch of the idea before we actually present the scheme.

Intuition. Any threshold attribute based signature ensures that the signer possesses atleast t out of the specified signing attributes, say n^* in number. Another way to look at this would be that, the signer has atleast 1 out of the $\binom{n^*}{t}$ combination of the attribute sets. Thus, a new approach to the same would be, for the signer to pick some n' sets of t attributes each from the $\binom{n^*}{t}$ possible sets, and prove that she has atleast one of the n' sets in her possession. Note here that $1 \leq n' \leq \binom{n^*}{t}$. If $n' \geq 2$, it would be sufficient to prove that the signature is valid and the signer has the specified attributes, moreover it would also give a reasonable degree of anonymity and not reveal the exact credentials of the signer. If the actual predicate is an AND, then $t = n^*$, which means $n' = 1$ and the signer can prove the possession of the complete set of attributes. On the other hand, if the predicate is a simple OR, then $t = 1$ and again the signer can choose an appropriate n' depending on the amount of privacy she wishes to have and then produce a signature. With this intuition, we are ready to see the details of the scheme.

3.1 Construction

We present the construction for our scheme which is based on the ring signature proposed in [4]. Here, for each set of attributes in the chosen n', we aggregate the attributes by summing them up and form n' components, one for each set. One of these components has the signer's private key embedded in it, making it a ring signature. During the verification phase the signer's component also takes care of eliminating all the attribute sets except the one which is actually used for signing, thus proving the possession of one among the chosen n' attribute sets. Our construction also allows the key-generating authority to revoke anonymity if required.

3.1.1 Setup

Let $U = \{A_1, A_2, \cdots, A_n\}$ denote the universe of attributes (attributes are denoted as A_i). Let t denote the threshold that a user needs to satisfy and U^* denote the set of attributes in the predicate. If $|U^*| = n^*$ then, a user must have atleast t out of the n^* attributes to be able to produce a valid signature on a message. Let $\mathbb{G}_1$ denote a cyclic additive group of prime order p on which the bilinear function is efficiently computable. Let $e(\cdot,\cdot)$ be the bilinear function, $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. Let, H_1, H_2, H_3, and H_4 be four hash functions defined by, $H_1 : \{0,1\}^* \to \{0,1\}^*$, $H_2 : \{0,1\}^* \to \mathbb{G}_1$, $H_3 : \{0,1\}^* \to \mathbb{G}_1$, and $H_4 : \{0,1\}^* \to \mathbb{Z}_p^*$. Let the generator of the group be $P \in_R \mathbb{G}_1$, secret key be $\alpha \in_R \mathbb{Z}_p^*$, and denote $\gamma = e(P,P) \in \mathbb{G}_2$. The master secret key (msk) is α and $P_{pub} = \alpha P$. The public parameters of the system,

$$params = (e, \mathbb{G}_1, \mathbb{G}_2, H_1(\cdot), H_2(\cdot), H_3(\cdot), H_4(\cdot), P, P_{pub})$$

3.1.2 Key Generation

$D \leftarrow$ KeyGen(U_β, ID, msk). Let, U_β be the set of attributes that a user β has. Let D denote the set of keys given to the user. Say, $|U_\beta| = n_\beta$. Here, the attribute authority picks a $r_\beta \in_R \mathbb{Z}_p^*$ and then computes the following:

1. Set $\bar{D_0} = r_\beta P$ and $\bar{D_1} = r_\beta^{-1} P$
2. $\omega = H_1(U_\beta, ID)$
3. Choose $w \in_R Z_p^*$ and set $W = wcP$
4. Set $\bar{D_2} = r_\beta^{-1} W$
5. Then, $Q_i = H_2(A_i)$ and $D_i = r_\beta \cdot \alpha \cdot Q_i \quad (\forall A_i \in U_\beta)$

The private key returned is, $D = \left\{\{D_i\}_{i \in \{1,\cdots,n_\beta\}}, \bar{D_0}, \bar{D_1}, \bar{D_2}, \omega\right\}$.

Key verification

The user, who on receiving the private keys corresponding to his/her attributes can verify the keys as follows:

$$e(\bar{D_0}, \bar{D_1}) \stackrel{?}{=} \gamma$$
$$e(\bar{D_0}, \bar{D_2}) \stackrel{?}{=} e(P, H_3(H_1(U_\beta, ID)))$$
$$e(D_i, \bar{D_1}) \stackrel{?}{=} e(Q_i, P_{pub})$$

3.1.3 Sign

$\sigma \leftarrow \text{Sign}(U^*, t, D, m, params)$. The signer who possesses atleast t of the attributes in U^* must be able to produce a valid signature on a message m. Let T_β be the t-element subset of attributes of U^*, that the user chooses inorder to generate the signature. i.e $T_\beta \subseteq U^* \cap U_\beta$ such that, $|T_\beta| = t$. Let T be a collection of n' subsets of attributes from U^* such that each of these subsets has a cardinality of exactly t and no two of them are equivalent. Let us assume that the sets in T are indexed by values from 1 to n'; and T_i denotes the subsets of U^* that constitute T. So, the signer chooses $\{T_i\}_{i\in\{1,\cdots,n'\}}$ where $T_i \subseteq U^*$, $|T_i| = t$, $T_i \neq T_j$ and $2 \leq n' \leq \binom{n^*}{t}$. Without loss of generality, we can assume that the set T_β is present in T and is at a random index s where $1 \leq s \leq n'$. Thus, we refer to T_β as T_s in our subsequent discussions. Hence, T_s is the set used for generating the signature, and the remaining $(n'-1)$ sets are used to form the ring (to provide anonymity). Then, the signer does the following computations to generate the signature:

The signer first picks n' random values, $r_i \in_R \mathbb{Z}_p^*$ for $i = \{1, \cdots, n'\}$, and an $r^* \in_R \mathbb{Z}_p^*$ and then proceeds to generate the signature on m as follows:

1. Set $\bar{V_0} = r^* \bar{D_0}, \quad \bar{V_1} = {r^*}^{-1} \bar{D_1} \quad \text{and} \quad \bar{V_2} = {r^*}^{-1} \bar{D_2}$
2. $U_i \in_R \mathbb{G}_1, \qquad \text{for } i = \{1, \cdots, n'\} \backslash s$
3. $h_i = H_4(m, U_i, T_i, \bar{V_0}, \bar{V_1}, \bar{V_2})$, for all $i \in \{1, \cdots, n'\} \backslash s$.
4. $U_s = \left(r_s \cdot \sum_{A_j \in T_s} Q_j \right) - \left(\sum_{i=1; i\neq s}^{n'} U_i + (h_i \sum_{A_j \in T_i} Q_j) \right)$
5. We define, $h_s = H_4(m, U_s, T_s, \bar{V_0}, \bar{V_1})$, in a manner consistent to that of the definition of the h_i values.
6. We set, $V = r^*(r_s + h_s) \sum_{A_i \in T_s} D_i$

The final signature σ is given as:

$$\sigma = \left\{ \{T_i\}_{i=\{1,\cdots,n'\}} \ , \ \{U_i\}_{i=\{1,\cdots,n'\}} \ , \ V \ , \ \bar{V_0} \ , \ \bar{V_1} \ , \ \bar{V_2}, \omega \right\}$$

It is important to note in the algorithm that the size of the signature is independent of the number of attributes, but depends more on the degree of privacy that the signer prefers. This is because, n' is a factor that the signer chooses, depending on the amount of information the signer wishes to reveal and does not depend on the size of U^* or t.

3.1.4 Verify

Any user can verify the signature by performing the following computations:

$$e(\bar{V_0} \ , \ \bar{V_1}) \stackrel{?}{=} \gamma$$
$$e(\bar{V_0} \ , \ \bar{V_2}) \stackrel{?}{=} e(P, H_3(\omega))$$
$$e(V \ , \ \bar{V_1}) \stackrel{?}{=} e\left(\sum_{i=1}^{n'} (U_i + h_i \sum_{A_j \in T_i} Q_j) \ , \ P_{pub} \right)$$

The signature is valid only if all the three checks are satisfied, in all other cases it's considered to be invalid. Correctness of the above three checks can be found in **Appendix(A)**.

4 Security

We will show that our threshold attribute based signature scheme is existentially unforgeable with respect to the *chosen attribute and message* attack as defined in Section 2.6. Then, we will also show that our scheme satisfies the weak signer attribute privacy property.

4.1 Unforgeability

Theorem 4.1 (Unforgeability) *In the random oracle model, if there exists an algorithm $\mathcal{A}$ that can win the existentially unforgeable chosen attribute and message attack game, with non-negligible probability ϵ, and create a valid ABS in polynomial time T, by making at most $q_{H_1}, q_{H_2}, q_{H_3}, q_{H_4}$ queries to the random oracles H_2, H_4, H_1, H_3, and, q_k and q_s Key-Generation and Sign queries respectively. Then the modified-computational bilinear Diffie-Hellman (m-CBDH) problem can be solved within expected time $\mathsf{T}' \leq \frac{144823(P_{Q,n}(\mathsf{T}+q_s\mathsf{T}_s)}{\epsilon}$ (where $Q = q_{H_1} + q_{H_2} + q_{H_3} + q_{H_4} + q_k$).*

Proof: The proof for the unforgeability (refer Theorem(4.1)) of our threshold attribute based signature follows, to some extent, that given by Chow *et al.* in [4]. In the subsequent discussion we will show the reduction of our scheme to solving the CBDH problem.

Inorder to solve the m-CBDH problem, the challenger $\mathcal{C}$ receives the instance $\{P, aP, bP, cP, a^{-1}P\}$ and has to finally produce $e(P,P)^{abc}$ as the output. The challenger will run $\mathcal{A}$ as a subroutine in the existential unforgeability game. As defined in the game, $\mathcal{A}$ can make queries to the hash functions; although the hash outputs will be random, the challenger $\mathcal{C}$ will maintain separate lists of the query and response of each oracle in order to simulate proper collision-resistant hash functions and avoid inconsistencies. Also, in the proof, we make the assumption that all the $H_2(A_i)$ queries are made before they are used in any further oracle queries.

Setting. First, the challenger $\mathcal{C}$ sets the system public-key as $P_{pub} = aP$ and master secret as $\alpha = a$. Note that $\mathcal{C}$ does not know a, b or c, but it will simulate those values during its responses to $\mathcal{A}$, with the help of aP, bP, cP and $a^{-1}P$.

H_1 **queries.** Queries to the H_1 oracle are answered by the challenger as follows. $\mathcal{C}$ picks an $\omega \in \mathbb{Z}_p^*$ uniformly at random, and then checks if this value is present in the list L_1. If it is present, then $\mathcal{C}$ re-picks ω repeating the process until it gets a new value. Then the tuple $< U_\beta, ID, \omega >$ is added to list L_1.

H_2 **queries.** When $\mathcal{A}$ makes queries to the hash function $H_2(\cdot)$ with input as some attribute A_i, $\mathcal{C}$ does the following. If A_i was already queried before, the hash value will be in the list L_2 and $\mathcal{C}$ will search and give the value stored in the list. Otherwise, $\mathcal{C}$ first picks an $s_i \in \mathbb{Z}_p^*$ uniformly at random, and then checks if this value is present in the list L_2. If it is present $\mathcal{C}$ re-picks s_i repeating the process until it gets a new value. Then, it sets $Q_i = H_2(A_i) = s_i(bP)$. After each response, $\mathcal{C}$ makes sure to save the tuple $< A_i, s_i, Q_i >$ in the list L_2, if it was not already present.

H_3 **queries.** List L_3 is used to maintain the queries and responses of this oracle. When an input ω is queried for its hash, the list L_3 is looked up to see if a matching entry already exists, if it is found the corresponding value is returned. In all other cases, an $w \in_R \mathbb{Z}_p^*$ is picked uniformly at random and W is set to be $W = wcP$. The tuple of, $< \omega, W, w >$ is added to the list, L_3.

Key-Gen queries. Adversary $\mathcal{A}$ is allowed to request for the private keys on any set of attributes U_β. When the challenger gets a query, $\mathcal{C}$ first picks an $r_\gamma \in_R \mathbb{Z}_p^*$, at random. Assume, $r_\beta = r_\gamma / a$. Then, the rest of the values are set as follows:

1. Set, $\bar{D_0} = r_\beta P = (r_\gamma a^{-1})P = r_\gamma(a^{-1}P)$ and $\bar{D_1} = r_\beta^{-1}P = (r_\gamma a^{-1})^{-1}P = (r_\gamma^{-1}a)P = r_\gamma^{-1}(aP)$
2. Now, $\omega = H_1(U_\beta, ID)$
3. $W = H_3(\omega)$, this is set as $W = wP$ $\qquad (w \in_R Z_p^*)$
4. Tuple $< U_\beta, ID, \omega >$ is added to L_1 and tuple $< \omega, w, W >$ is added to L_3.
5. Compute, $\bar{D_2} = r_\beta^{-1}W = r_\gamma^{-1}a(wP) = r_\gamma^{-1}w(aP)$
6. $\forall A_i \in U_\beta$, $D_i = r_\beta \cdot \alpha \cdot Q_i = r_\gamma a^{-1}a(s_i bP) = r_\gamma s_i(bP)$

The final key is given as, $D = \big\{\{D_i\}_{i\in\{1,\cdots,n_\beta\}}, \bar{D_0}, \bar{D_1}, \bar{D_2}, \omega\big\}$.

In each of the hash oracle queries, all the intermediate random values that have been chosen by the challenger are added to the respective lists along with the computed components and final responses.

H_4 **queries.** Whenever queries to $H_4(\cdot)$ are made, $\mathcal{C}$ first looks up entries in L_4 to see if the same query was made previously. If a matching entry is found, it gives the corresponding saved hash value, otherwise, it just picks a random value from $\mathbb{Z}_p^*$ and gives it as output, storing the input and response as a tuple in L_4.

Sign. Signature requests are answered by $\mathcal{C}$ as follows:

It picks a T_s at random first. Then selects n' random values, $r_i \in_R \mathbb{Z}_p^*$ for $i = \{1, \cdots, n'\}$, and computes the following:

1. $V_0 = (r^*)^{-1}\bar{D_0} = (r^*)^{-1}(r_{\beta'})P$,
   $\bar{V_1} = (r^*)^{-1}\bar{D_1} = (r^* r_{\beta'})^{-1}P$,
   $\bar{V_2} = (r^*)^{-1}\bar{D_2} = (r^* r_{\beta'})^{-1}H_3(\omega)$
2. Add the values $r_\beta, U_\beta, \bar{D_0}, \bar{D_1}, \bar{D_2}, T_s$ to the sign oracle list.
3. Picks $z \in_R \mathbb{Z}_p^*$ and sets[1], $V = r^* r_{\beta'} \cdot z \cdot (aP)$
4. For $i \in \{1, \cdots, n'\} \backslash s$
   - Selects $U_i \in_R \mathbb{G}_1$
   - Gets $h_i = H_4(m, U_i, T_i, \bar{V_0}, \bar{V_1}, \bar{V_2})$ and saves h_i in list L_4
5. $U_s = zP - \left(h'_s \cdot \sum_{A_j \in T_s} Q_j \right) - \left(\sum_{i \neq s} U_i + (h_i \sum_{A_j \in T_i} Q_j) \right)$
6. $V = r^{*-1} r_{\beta'} z(aP)$
7. Save the tuple $< h'_s, U_s, T_s, \bar{V_0}, \bar{V_1}, \bar{V_2}, V >$ in L_4.
8. $\sigma = \left\{ \{T_i\}_{i=\{1,\cdots,n'\}} , \{U_i\}_{i=\{1,\cdots,n'\}} , V , \bar{V_0} , \bar{V_1} , \bar{V_2} , \omega \right\}$

Proof for the correctness of this signature can be found in **Appendix (B)**.

Forgery. Finally, $\mathcal{A}$ will output $\sigma = \left\{ \{T_i, U_i\}_{i=\{1,\cdots,n'\}} , V , \bar{V_0} , \bar{V_1} , \bar{V_2} , \omega \right\}$, the forged signature on the message m.

Solving CBDH. From the forking lemma for generic ring signature schemes [7] it follows that, if with non-negligible probability, $\mathcal{A}$ can give a valid forged signature in the above interaction within time $T_{\mathcal{A}}$, then we can construct another algorithm $\mathcal{A}'$ which within time $2T_{\mathcal{A}}$ can output two signatures, $\sigma = \left\{ \{T_i, U_i\}_{i=\{1,\cdots,n'\}}, V, \bar{V_0}, \bar{V_1}, \bar{V_2}, \omega \right\}$; and $\sigma' = \left\{ \{T_i, U_i\}_{i=\{1,\cdots,n'\}}, V', \bar{V_0'}, \bar{V_1'}, \bar{V_2'}, \omega \right\}$ also with non-negligible probability. It also follows from the lemma that with non-negligible probability, we can have $h_i = h'_i$, for all $i \in \{1, \cdots, n'\} \backslash s$. Using this $\mathcal{C}$ can solve for $e(P,P)^{abc}$ as:

$$\left[\left(\frac{e(V, \bar{V_2})}{e(V', \bar{V_2'})} \right)^{w^{-1}(h_s - h'_s)^{-1}} \right]^{(\sum_{A_i \in T_s} s_i)^{-1}}$$

Now, given $\mathcal{A}'$ derived from $\mathcal{A}$, $\mathcal{C}$ can solve for $e(P,P)^{abc}$.

A more detailed derivation, showing the computations involved in extracting the solution for CBDH can be found in the **Appendix(B.1)**. ■

4.2 Signer Attribute Privacy

In this section we will define what privacy is and prove that our scheme provides privacy(or anonymity) to the signer's attribute subset used in the signature.

We define signer ambiguity for our scheme in a manner similar to the one given in [4] for ring signatures. An attribute-based signature scheme, using the ring approach as defined by us, for the threshold access structure, is said to have unconditional signer attribute-set ambiguity if for any group of n' attribute subsets $\{T\}$, where $T = \bigcup T_i$, $\forall 1 \leq i \leq n'$, $T_i \subseteq U^*$ and $|T_i| = t$, any message m and any signature σ, where $\sigma = Sign(m, t, U^*)$; any verifier $\mathcal{A}$ even with unbounded computing resources, cannot identify the actual attribute subset of the signer (used in the signature) with probability better than a random guess. That is, $\mathcal{A}$ can output the actual signer's chosen attribute subset (indexed by T_s) with probability no better than $1/n'$.

Theorem 4.2 (Weak signer attribute privacy) *Our threshold attribute-based signature has weak signer attribute privacy property.*

[1]Note: This cannot be done by a normal signer since $r_{\beta'}$ will only be available to the attribute authority.

Proof: We first claim that all the U_i's are uniformly distributed. This is because, each U_i (including U_s) is obtained via multiplying the components with a value r_i, that is chosen uniformly at random. So, we can say that the U_i's by themselves (as independent entities) don't leak any information. Another component of the signature, ω, is a hash of the attributes of the user, but since it's a hash and is created even before T_s is chosen, it cannot reveal anything about T_s. Also, the other values, $\bar{V}_0, \bar{V}_1$ and $\bar{V}_2$ are unrelated to T_s. So, it remains to be seen if V gives away any information about T_s with the help of the bilinear map function along with any of the given components and public values.

So, we will consider if $V = r^*(r_s + h_s)\sum_{A_i \in T_s} D_i$, leaks anything about T_s. Let us focus on $V - r^* h_s \sum_{A_i \in T_s} D_i = r^* r_s \sum_{A_i \in T_s} D_i$. The h_s component can be obtained publicly since it is a hash. We'll see if this component gives away information related to T_s when considered along with $\bar{V}_1 = r^{*-1} r_\beta^{-1} P$, in the bilinear map. If we manage to get $r^* r_s \sum_{A_i \in T_s} D_i$, then we can do the following verification test: check $e(r^* r_s \sum_{A_i \in T_s} D_i, \bar{V}_1) = e(r_s \sum_{A_i \in T_s} Q_i, P_{pub})$? To do this, any user who suspects that the set T_k was used in signing of the message will only need to check if, $e(U_k + \sum_{i \neq k}(U_i + h_i \sum_{A_j \in T_i} Q_j), P_{pub}) \stackrel{?}{=} e(V, \bar{V}_1)/e(h_k \sum_{A_j \in T_k} Q_j, P_{pub})$.

We will now show that, although the above equality is valid for $k = s$, it is equally valid for any of the other attribute subsets in T i.e the check is symmetric with respect to any attribute subset and hence does not reveal anything about T_s. To see that, consider:

$$
\begin{aligned}
U_k + \sum_{i \neq k}(U_i + h_i \sum_{A_j \in T_i} Q_j) &= U_s + \textstyle\sum_{i \neq s}(U_i) + \sum_{i \neq k}(h_i \sum_{A_j \in T_i} Q_j \\
&= \left(r_s \cdot \sum_{A_j \in T_s} Q_j \right) - \left(\sum_{i \neq s} U_i + (h_i \sum_{A_j \in T_i} Q_j) \right) \\
&\quad + \sum_{i \neq s}(U_i) + \sum_{i \neq k}(h_i \sum_{A_j \in T_i} Q_j) \\
&= r_s \cdot \sum_{A_j \in T_s} Q_j - h_k \sum_{A_j \in T_k} Q_j + h_s \sum_{A_j \in T_s} Q_j \\
&= (V)^{r^{*-1} r_\beta^{-1} \alpha^{-1}} - h_k \sum_{A_j \in T_k} Q_j
\end{aligned}
$$

Thus,

$$
\begin{aligned}
e(U_k + \sum_{i \neq k}(U_i + h_i \sum_{A_j \in T_i} Q_j), P_{pub}) \\
= e(V, \bar{V}_1)/e(h_k \sum_{A_j \in T_k} Q_j, P_{pub})
\end{aligned}
$$

This proves that the check is symmetric with all attribute subsets in $T = \bigcup T_i$, $\forall 1 \leq i \leq n'$. So, the signature components are independent and uniformly distributed irrespective of the attribute subset being used. Thus, our scheme is signer attribute-set anonymous and hence satisfies weak signer atribute privacy. ■

5 Conclusion

5.1 Advantages of the new approach.

Controlled privacy. The proposed threshold scheme has a new property that we can call controlled attribute privacy which is not known to be present (to the best of the our knowledge) in any of the previous threshold attribute based signature schemes. This is a feature that would allow the signers to control the privacy/anonymity of therir attributes even if the signing policy is not determined by them. We will illustrate this feature with an example. Let us say Alice is signing a document which wants the signer to satisfy a threshold predicate, and she has sufficient attributes to satisfy the predicate. Say, one of the attributes of the signing policy is *CIA officer*. Now, Alice being a *CIA officer* among other things wishes to highlight this particular fact in her signature (although it may not be necessary). She can choose all the n' attribute sets $\{T_i\}_{i \in \{1, \cdots, n'\}}$ with *CIA officer* being one of the attributes in each of these sets. By doing this, she has control over which of her attributes she wants to reveal. But if Alice does not wish to reveal anything about her credentials except that they satisfy the necessary threshold, then she will have to give all the $\binom{n^*}{t}$ possible sets. If on the other-hand, Alice is completely indifferent about revealing all of her attributes, then she can give a signature and include a single subset of attributes. And that set should contain just the exact set of attributes used in the signature inorder to satisfy the given policy. Note that this will also be a constant size signature, since it will have only one T_i and U_i.

Size-Privacy balance. The power that this feature gives is that, even if the signing policy is specified by a different authority, the signer can choose to reveal more in the signature than what other schemes would normally allow. In a way, our approach allows the signer control over the signature size and privacy, although he/she may not have had the freedom to set the signing policy. If a signer does not care about privacy, then she can go for a constant size signature. On the other-hand if the size of the signature components is immaterial, then signer can choose to get complete privacy by choosing all the subsets of attributes satisfying the policy to be a part of the signature.

Multi-level threshold attribute based signature. We believe that this scheme can be extended to a multi-level threshold attribute based signature provided each attribute is present only once in the predicate. However, it would be interesting to see if it can extended for a general multi-level threshold ABS.

5.2 Summary.

In this work we presented a new approach to t-ABS based on ring signatures. We have also given a scheme and shown it to be existentially unforgeable with respect to *chosen attribute and message* attack using the random oracle model. In addition, the scheme has also been proved to provide the weak signer attribute privacy property. We believe that this new approach to t-ABS gives the signer greater power to control his/her anonymity even if (s)he does not get to determine the signing policy.

References

[1] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, pages 321–334, 2007.

[2] Xavier Boyen. Mesh signatures. In *Proceedings of the 26th annual international conference on Advances in Cryptology*, EUROCRYPT '07, pages 210–227, Berlin, Heidelberg, 2007. Springer-Verlag.

[3] Jan Camenisch. Efficient and generalized group signatures. In *Proceedings of the 16th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'97, pages 465–479, Berlin, Heidelberg, 1997. Springer-Verlag.

[4] Sherman S. M. Chow, S. M. Yiu, and Lucas C. K. Hui. Efficient identity based ring signature. In *Applied Crypto and Network Security - ACNS 2005, LNCS 3531*, pages 499–512. Springer, 2005.

[5] Sherman Chow Sze Ming. Forward security from bilinear pairings: Signcryption and threshold signature. Master's thesis, University of Hong Kong, August 2004.

[6] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, CCS '06, pages 89–98, New York, NY, USA, 2006. ACM.

[7] Javier Herranz and Germán Sáez. New identity-based ring signature schemes. In *ICICS'04*, pages 27–39, 2004.

[8] Dalia Khader. Attribute based group signature with revocation. Cryptology ePrint Archive, Report 2007/241, 2007. `http://eprint.iacr.org/`.

[9] Dalia Khader. Attribute based group signatures. Cryptology ePrint Archive, Report 2007/159, 2007. `http://eprint.iacr.org/`.

[10] Swarun Kumar, Shivank Agrawal, Subha Balaraman, and C Pandu Rangan. Attribute based signatures for bounded multi-level threshold circuits. In *Proceedings of the 7th European Workshop on Public Key Services, Applications and Infrastructures*, EuroPKI '10, 2010 (to appear).

[11] Jin Li, Man Ho Au, Willy Susilo, Dongqing Xie, and Kui Ren. Attribute-based signature and its applications. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '10, pages 60–69, New York, NY, USA, 2010. ACM.

[12] Jin Li and Kwangjo Kim. Attribute-based ring signatures. Cryptology ePrint Archive, Report 2008/394, 2008. `http://eprint.iacr.org/`.

[13] Hemanta Maji, Manoj Prabhakaran, and Mike Rosulek. Attribute-based signatures: Achieving attribute-privacy and collusion-resistance. Cryptology ePrint Archive, Report 2008/328, 2008. `http://eprint.iacr.org/`.

[14] Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. Attribute-based signatures. Cryptology ePrint Archive, Report 2010/595, 2010. http://eprint.iacr.org/.

[15] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, pages 554–567. Springer-Verlag, 2001.

[16] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pages 457–473, 2005.

[17] Siamak F. Shahandashti and Reihaneh Safavi-Naini. Threshold attribute-based signatures and their application to anonymous credential systems. In *Proceedings of the 2nd International Conference on Cryptology in Africa: Progress in Cryptology*, AFRICACRYPT '09, pages 198–216, Berlin, Heidelberg, 2009. Springer-Verlag.

[18] Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. Cryptology ePrint Archive, Report 2008/290, 2008. http://eprint.iacr.org/.

Appendix

A New scheme's Sign algorithm correctness

We argue here that, if the steps of the algorithm are followed without deviation then, the signature given is valid. We will show a proof of the correctness mathematically. Let's first consider $e(\bar{V_0}\ ,\ \bar{V_1})$:

$$\begin{aligned} e(\bar{V_0}\ ,\ \bar{V_1}) &= e(r^*\bar{D_0}\ ,\ r^{*-1}\bar{D_1}) \\ &= e(r^* r_\beta P\ ,\ r^{*-1} r_\beta^{-1} P) \\ &= e(P\ ,\ P) \\ &= \gamma \end{aligned}$$

Next we check $e(\bar{V_0}\ ,\ \bar{V_2})$:

$$\begin{aligned} e(\bar{V_0}\ ,\ \bar{V_2}) &= e(r^*\bar{D_0}\ ,\ r^{*-1}\bar{D_2}) \\ &= e(r^* r_\beta P\ ,\ r^{*-1} r_\beta^{-1} H_3(\omega)) \\ &= e(P\ ,\ H_3(\omega)) \end{aligned}$$

Now, we will see if the third verification is also valid. i.e Check if,
$e(V\ ,\ \bar{V_1}) = e\left(\sum_{i=1}^{n'}(U_i + h_i \sum_{A_j \in T_i} Q_j)\ ,\ P_{pub}\right)$ is correct for a valid signature. We look at the L.H.S and the R.H.S components separately in showing the proof.

Consider the L.H.S:

$$\begin{aligned} e\left(V\ ,\ \bar{V_1}\right) &= e\left(r^*(r_s + h_s) \sum_{A_i \in T_s} D_i\ ,\ r^{*-1}\bar{D_1}\right) \\ &= e\left((r_s + h_s) \sum_{A_i \in T_s} D_i\ ,\ r_\beta^{-1} P\right) \\ &= e\left((r_s + h_s) \cdot r_\beta \cdot \alpha \sum_{A_i \in T_s} Q_i\ ,\ r_\beta^{-1} P\right) \\ &= e\left((r_s + h_s) \cdot \sum_{A_i \in T_s} Q_i\ ,\ P_{pub}\right) \end{aligned} \tag{1}$$

Now, for the R.H.S:

$$e\left(\sum_{i=1}^{n'}(U_i + h_i \sum_{A_j \in T_i} Q_j)\ ,\ P_{pub}\right)$$

We'll consider the first component $\sum_{i=1}^{n'}(U_i + h_i \sum_{A_j \in T_i} Q_j)$ and simplify it before we compute the mapping.

$$\sum_{i=1}^{n'}(U_i + h_i \sum_{A_j \in T_i} Q_j) = \sum_{i=1}^{n'} U_i + \sum_{i=1}^{n'}(h_i \cdot \sum_{A_j \in T_i} Q_j)$$

$$= U_s + \sum_{i=1;i\neq s}^{n'} U_i + (h_s \cdot \sum_{A_j \in T_s} Q_j) + \sum_{i=1;i\neq s}^{n'} (h_i \cdot \sum_{A_j \in T_i} Q_j)$$

$$= \left(r_s \cdot \sum_{A_j \in T_s} Q_j\right) - \left(\sum_{i\neq s} U_i + (h_i \sum_{A_j \in T_i} Q_j)\right)$$

$$+ (h_s \cdot \sum_{A_j \in T_s} Q_j) + \sum_{i=1;i\neq s}^{n'} U_i + \sum_{i=1;i\neq s}^{n'} (h_i \cdot \sum_{A_j \in T_i} Q_j)$$

$$= \left(r_s \cdot \sum_{A_j \in T_s} Q_j\right) - \sum_{i=1;i\neq s}^{n'} U_i - \sum_{i=1;i\neq s}^{n'} (h_i \cdot \sum_{A_j \in T_i} Q_j)$$

$$+ (h_s \cdot \sum_{A_j \in T_s} Q_j) + \sum_{i=1;i\neq s}^{n'} U_i + \sum_{i=1;i\neq s}^{n'} (h_i \cdot \sum_{A_j \in T_i} Q_j)$$

$$= (r_s + h_s) \cdot \sum_{A_i \in T_s} Q_i \tag{2}$$

Using the above we get R.H.S to be,

$$e\left(\sum_{i=1}^{n'}(U_i + h_i \sum_{A_j \in T_i} Q_j)\ ,\ P_{pub}\right) = e\left((r_s + h_s) \cdot \sum_{A_i \in T_s} Q_i\ ,\ P_{pub}\right) \tag{3}$$

Thus, from equations (1), (2) and (3) we can see that the verification holds and can be performed using the public values.

B Sign Oracle Correctness

We will show the proof for the verification of the signature generated by the oracle while showing the security of the scheme (from Section 4.1).
The signature components generated by the sign oracle are as follows:

1. $V_0 = r^* \bar{D_0} = r^{*-1} r_{\beta'} P, \qquad \bar{V_1} = r^{*-1} \bar{D_1} = r^{*-1} r_{\beta'}^{-1} P$
   $\bar{V_2} = r^{*-1} \bar{D_2} = r^{*-1} r_{\beta'}^{-1} H_3(\omega)$
2. $V = r^* r_{\beta'} \cdot z \cdot (aP)$
3. For $i \in \{1, \cdots, n'\} \backslash s$
   - Sets $U_i \in_R \mathbb{G}_1$
   - Gets $h_i = H_4(m, U_i, T_i, \bar{V_0}, \bar{V_1}, \bar{V_2})$
4. $U_s = zP - \left(h'_s \cdot \sum_{A_j \in T_s} Q_j\right) - \left(\sum_{i\neq s} U_i + (h_i \sum_{A_j \in T_i} Q_j)\right)$
5. $V = r^{*-1} r_{\beta'} z(aP)$
6. $\sigma = \left\{\{T_i\}_{i=\{1,\cdots,n'\}}\ ,\ \{U_i\}_{i=\{1,\cdots,n'\}}\ ,\ V\ ,\ \bar{V_0}\ ,\ \bar{V_1}\ ,\ \bar{V_2},\ ,\ \omega\right\}$

Now, the verification has to satisfy the following three equations:

$$e(\bar{V_0}\ ,\ \bar{V_1}) \stackrel{?}{=} \gamma$$

$$e(\bar{V_0}\ ,\ \bar{V_2}) \stackrel{?}{=} e(P, H_3(\omega))$$

$$e(V\ ,\ \bar{V_1}) \stackrel{?}{=} e\left(\sum_{i=1}^{n'}(U_i + h_i \sum_{A_j \in T_i} Q_j)\ ,\ P_{pub}\right)$$

B.1 Verification analysis

Let's first consider $e(\bar{V_0}\ ,\ \bar{V_1})$:

$$e(\bar{V_0}\ ,\ \bar{V_1}) = e(\bar{D_0}\ ,\ \bar{D_1}) = e(r_\beta P\ ,\ r_\beta^{-1}P) = e(P\ ,\ P) = \gamma$$

Similarly, for $e(\bar{V_0}\ ,\ \bar{V_2})$:

$$e(\bar{V_0}\ ,\ \bar{V_2}) = e(\bar{D_0}\ ,\ \bar{D_2}) = e(r^* r_\beta P\ ,\ r^{*-1} r_\beta^{-1} H_3(\omega)) = e(P\ ,\ H_3(\omega))$$

Now, we will see if $e(V\ ,\ \bar{V_1}) = e\left(\sum_{i=1}^{n'}(U_i + h_i \sum_{A_j \in T_i} Q_j)\ ,\ P_{pub}\right)$ will hold true.

Consider the L.H.S:

$$\begin{aligned} e\left(V\ ,\ \bar{V_1}\right) &= e\left(r_{\beta'} \cdot z \cdot (aP)\ ,\ r_\beta^{-1} P\right) \\ &= e\left(z \cdot (aP)\ ,\ P\right) \\ &= e\left(zP\ ,\ P_{pub}\right) \end{aligned} \tag{4}$$

Now, for the R.H.S:

$$e\left(\sum_{i=1}^{n'}(U_i + h_i \sum_{A_j \in T_i} Q_j)\ ,\ P_{pub}\right)$$

Let's just consider the first component:

$$\begin{aligned} \sum_{i=1}^{n'}(U_i + h_i \sum_{A_j \in T_i} Q_j) &= \sum_{i=1}^{n'} U_i + \sum_{i=1}^{n'}(h_i \cdot \sum_{A_j \in T_i} Q_j) \\ &= U_s + \sum_{i=1;i\neq s}^{n'} U_i + (h'_s \cdot \sum_{A_j \in T_s} Q_j) + \sum_{i=1;i\neq s}^{n'} (h_i \cdot \sum_{A_j \in T_i} Q_j) \\ &= zP - \left(h'_s \cdot \sum_{A_j \in T_s} Q_j\right) - \left(\sum_{i=1;i\neq s}^{n'} U_i + (h_i \sum_{A_j \in T_i} Q_j)\right) \\ &\qquad + (h'_s \cdot \sum_{A_j \in T_s} Q_j) + \sum_{i=1;i\neq s}^{n'} U_i + \sum_{i=1;i\neq s}^{n'} (h_i \cdot \sum_{A_j \in T_i} Q_j) \\ &= zP \end{aligned} \tag{5}$$

Thus, R.H.S also reduces to,

$$e\left(\sum_{i=1}^{n'}(U_i + h_i \sum_{A_j \in T_i} Q_j)\ ,\ P_{pub}\right) = e(zP, P_{pub}) \tag{6}$$

From equations (4), (5) and (6) we can see that the verification holds for the constructed signature.

Correctness of Solving CBDH After using the forking lemma (refer Section 4.1), let us say we have two signatures σ and σ' which have the following components:

$$V = r_1^*(r_s + h_s) \sum_{A_i \in T_s} D_i \qquad V' = r_2^*(r_s + h'_s) \sum_{A_i \in T_s} D_i$$

$$\bar{V_2} = r_1^{*-1} {r_{\beta 1}}^{-1} wcP \qquad \bar{V'}_2 = r_2^{*-1} {r_{\beta 2}}^{-1} wcP$$

$$\begin{aligned} V &= r^* r_{\beta 1}(r_s + h_s)(a \sum_{A_i \in T_S} Q_i) \\ &= r^* r_{\beta 1}(r_s + h_s)(ab(\sum_{A_i \in T_S} s_i)P) \\ W_1 &= e(V, \bar{V_2}) \\ &= e(P, P)^{(r_s + h_s)(ab(\sum_{A_i \in T_S} s_i))(wc)} \\ X_1 &= W_1^{w^{-1}} = \gamma^{(r_s + h_s)(abc \sum s_i)} \end{aligned} \tag{7}$$

Similarly, set $W_2 = e(V', \bar{V'}_2)$ and get X_2 as follows:

$$X_2 = W_2^{w^{-1}} = \gamma^{(r_s + h'_s)(abc \sum s_i)}$$

Now, we do the following,

$$\begin{aligned} Y_1 &= \frac{X_1}{X_2} = \gamma^{(h_s - h'_s)(abc \sum s_i)} \\ Y &= Y_1^{(h_s - h'_s)^{-1}} = \gamma^{(abc \sum s_i)} \\ Z &= (Y)^{(\sum s_i)^{-1}} = \gamma^{abc} = e(P, P)^{abc} \end{aligned} \tag{8}$$